

# 第十一章

## 文档布局及相互联系

潘建瑜

华东师范大学数学系

2010.05

# 本讲内容

- 1 document选项
- 2 页面版式
- 3 自定义页眉与页脚
- 4 目录表与图表目录
- 5 文档的分割处理
- 6 交叉引用与索引

# document选项

`\documentclass[可选项]{文档类别}`

- 基本字体尺寸: 10pt, 11pt, 12pt
- 纸张大小: letterpaper, a4paper, a5paper, ...
- 排版方向: portrait (竖向), landscape (横向)
- 是否生成独立的标题页: titlepage, notitlepage
- final 和 draft

# document选项

`\documentclass[可选项]{文档类别}`

- 基本字体尺寸: 10pt, 11pt, 12pt
- 纸张大小: letterpaper, a4paper, a5paper, ...
- 排版方向: portrait (竖向), landscape (横向)
- 是否生成独立的标题页: titlepage, notitlepage
- final 和 draft

---

👉 分栏: onecolumn, twocolumn

👉 单双面选项: oneside, twoside

👉 公式相关: leqno, fleqn, mathindent

# 分栏

## 分栏

```
\documentclass[twocolumn]{article}
```

☞ 缺省为 `onecolumn`

☞ 选用 `twocolumn` 时, 可指定

- 栏间距: `\columnsep`  $\leftarrow$  `\setlength`
- 栏间分隔线粗细: `\columnseprule`, 缺省为 0

☞ 栏宽参数: `columnwidth`

- 通常不用设置, 但可以使用

☞ 若对整篇文档作修改, 可将设置命令放在导言区

# 分栏

## □ 部分文档分栏

`\twocolumn`[通栏文本]

☞ `\twocolumn`

→ 另起新页, 以两栏格式排版其后面的文本

☞ `\onecolumn` → 新起一页, 返回单栏格式

☞ 双栏排版时, 先排满左栏, 再排右栏

☞ 例 1102.tex

# 分栏

## ❏ flushend 和 cuted 宏包

👉 flushend 宏包: 使得选用 twocolumn 版式时, 双栏的最后一页上的左右两栏基本等高

- \raggedend → 取消平衡最后一页双栏的功能
- \flushend → 恢复平衡最后一页双栏的功能

👉 cuted 宏包 → 在双栏版式中插入单栏文本

- 将单栏文本放入 strip 环境中
- cuted 宏包与 \onecolumn 的区别: 不另起新页

👉 此时使用 \onecolumn 和 \twocolumn 命令, 仍另起新页

# 分栏

## ❑ 多栏排版宏包: multicol

```
\usepackage{multicol}
...
\begin{multicols}{栏数}
...
\end{multicols}
```

👉 详见课程主页的宏包指南 (multicol.pdf)

👉 例 1102.tex



# 单双面

## ❑ 单双面选项: `oneside`, `twoside`

```
\documentclass[oneside]{article}  
\documentclass[twoside]{article}
```

👉 `article` 缺省是 `oneside`

`book` 缺省为 `twoside`

👉 `book` 类文档选项: `openany`, `openright`  
控制每章的开始位置, 缺省为 `openright`.

👉 单面版式: 所有页面均称为 **右页**

双面版式: 奇数页为 **右页**, 偶数页为 **左页**

# 单双面

## ❑ 单双面选项: `oneside`, `twoside`

☞ 单面版式: 所有页码打印位置一致

双面版式: 页码打印在外侧 → 奇右偶左

☞ 单面版式: 左边距相同, 右边距相同

双面版式: 内侧边距相同, 外侧边距相同

☞ 双面版式:

每页正文底线位置在相同位置上 (`flushbottom`)

☞ 单面版式:

每页正文底线位置可稍有不同 (`raggedbottom`)

# 与公式有关的选项

## 与公式有关的选项

```
\documentclass[leqno,fleqn]{article}
```

👉 **leqno**: 公式编号位于左边

👉 **fleqn**: 行间公式左对齐

👉 **mathindent**: 行间公式左对齐时的缩进距离

1 document选项

2 页面版式

3 自定义页眉与页脚

4 目录表与图表目录

5 文档的分割处理

6 交叉引用与索引

# 页面布局

## ❑ 页面上排放文本区域的四个部分

- ☞ 页眉 (header)
- ☞ 页芯 (body)
- ☞ 页脚 (footer)
- ☞ 边注 (margin notes)

## ❑ 页面布局图 (见教材或课程主页)

- ☞ `\textwidth`, `\textheight`
- ☞ `\hoffset`, `\voffset`
- ☞ `\oddsidemargin`, `\evensidemargin`

# 控制页面版式

## ❑ 控制页面版式的命令

```
\pagestyle{版式}
```

## ❑ 常用版式

- 👉 **plain** → 缺省选项, 页眉为空, 页码居中
- 👉 **empty** → 页眉页脚都为空, 不显示页码
- 👉 **headings** → 显示页眉 (含页码, 页眉标题)
  - 页眉标题由章, 节, 或小节标题组成, 自动提取
  - **book** 每一章的第一页和 **article** 的首页不显示页眉, 但在底部显示页码
  - 例: 1103h1.tex, 1103hbook.tex

# 控制页面版式

## • headings 版式中页眉标题的组成

文档类		左页	右页
book, report	单面格式	——	章
	双面格式	章	节
article	单面格式	——	节
	双面格式	节	小节

👉 `myheadings` → 与 `headings` 类似, 但可以使用 `\markright` 或 `\markboth` 来定义页眉标题

👉 例: 1103h2.tex

# 定义页眉标题

`\markright{右页页眉标题}`  
`\markboth{左页页眉标题}{右页页眉标题}`

- ☞ 单页版式中使用 `\markright`
- ☞ 双页版式中使用 `\markboth`
- ☞ 缺省使用章节标题作为页眉标题
- ☞ `headings` 版式中也可以使用这两个命令
- ☞ `\leftmark` → 存放左页页眉标题
- ☞ `\rightmark` → 存放右页页眉标题
- ☞ 改变当前页的页面版式: `\thispagestyle{版式}`



# 页码

## 指定页码样式

`\pagenumbering{数字样式}`

☞ `arabic`, `roman`, `Roman`, `alph`, `Alph`,

☞ 页码计数器: `page`

☞ 每调用一次 `\pagenumbering`, 计数器 `page` 置零

☞ `book` 类中三个控制版式命令:

- `\frontmatter` → 通常放在前言和目录前
- `\mainmatter` → 通常放在正文主体前
- `\backmatter` → 通常放在参考文献和索引前

- 1 document选项
- 2 页面版式
- 3 自定义页眉与页脚
- 4 目录表与图表目录
- 5 文档的分割处理
- 6 交叉引用与索引

# 自定义页眉与页脚

## 自定义页眉与页脚的宏包: fancyhdr

```
\usepackage{fancyhdr}  
\pagestyle{fancy}
```

- 该宏包将页面分为奇数页和偶数页
- 页眉和页脚都被分为三个区域: 左区, 中区, 右区
  - 左区靠左边界, 中区居中, 右区靠右边界
- 各区内容不能太长, 否则会出现文本重叠

# 自定义页眉与页脚

## 自定义页眉与页脚的命令:

```
\fancyhead[选项]{内容}
```

```
\fancyfoot[选项]{内容}
```

☞ **选项** 为字母 L, C, R (代表区域) 和 O, E (代表奇偶页) 的组合

- 若 **选项** 中不含 **区域字母**, 则表示所有区域  
若 **选项** 中不含 **奇偶页字母**, 则表示所有页面

☞ **简化的页眉页脚命令:**

- `\lhead`, `\chead`, `\rhead`
- `\lfoot`, `\cfoot`, `\rfoot`

# 自定义页眉与页脚

## ❑ 页眉 (页脚) 与正文之间分隔线的粗细:

```
\headrulewidth  
\footrulewidth
```

👉 缺省值分别为 0.4pt 和 0pt

👉 修改它们的值要用 `\renewcommand` 命令!

👉 例: 1103hfancy.tex

## ❑ 更一般的页眉页脚命令: `\fancyhf`

👉 `\fancyhead` → 带 H 选项的 `\fancyhf`

👉 `\fancyfoot` → 带 F 选项的 `\fancyhf`

# 自定义页眉与页脚

## ❑ 首页显示页眉

在 `\maketitle` 后使用 `\thispagestyle{fancy}`

## ❑ 重新定义 plain 样式

👉 一些  $\text{\LaTeX}$  命令, 如 `\chapter`, `\maketitle` 等, 使用 `\thispagestyle` 来切换到 `plain` 样式

👉 `\fancypagestyle` 可重新定义现有页面样式或创建新的页面样式

`\fancypagestyle{页面样式名称}{页面格式}`

# 自定义页眉与页脚

## ❑ 使用章节标题作页眉

- 👉 重新定义命令 `\chaptermark`, `\sectionmark` 和 `\subsectionmark`
- 👉 `\leftmark` 和 `\rightmark`

## ❑ 用一行显示章的标题

- 例: 1103hfancybook.tex

- 1 document选项
- 2 页面版式
- 3 自定义页眉与页脚
- 4 目录表与图表目录**
- 5 文档的分割处理
- 6 交叉引用与索引



# 目录表

## ❑ 自动生成目录

```
\tableofcontents
```

👉 设置出现在目录中的章节层次

```
\setcounter{tocdepth}{数}
```

- 数 → 代表章节层次
- 为了生产正确的目录, 通常需编译多次
- 只有自动编号的章节才会在目录中显示  
(即, 带星号的章节环境不在目录中输出)

# 目录表

👉 在目录中输出不编号的章节

```
\addcontentsline{toc}{章节名称}{条目内容}  
\addtocontents{toc}{条目内容}
```

- 章节名称 → part, chapter, section, ...

👉 不同的章节层次采用不同的格式

👉 第一个命令按章节名称的格式输出条目内容

第二个命令按原样输出条目内容, 无页码

# 插图目录与表格目录

## 插图目录与表格目录

`\listoffigures`, `\listoftables`

- 输出浮动环境 `figure` 和 `table` 生成的图表目录
- 图表目录中显示的为 `\caption` 中的内容
- 手工插入其它图表目录

`\addcontentsline{清单类型}{格式}{条目内容}`  
`\addtocontents{清单类型}{格式}`

清单类型: `toc` (目录), `lof` (插图), `lot` (表格)

格式: 章节名称, `figure`, `table`

# 目录清单标题中文化

## ❑ 目录清单标题中文化

```
\renewcommand{\contentsname}{\hei 目\quad 录}
```

```
\renewcommand{\listfigurename}{\hei 插图目录}
```

```
\renewcommand{\listtablename}{\hei 表格目录}
```

1 document选项

2 页面版式

3 自定义页眉与页脚

4 目录表与图表目录

5 文档的分割处理

6 交叉引用与索引

# 文档的分割处理

❑ 文档的分割处理: 将大文件分成几个小文件

❑ `\input` 命令: `\input{文件名}`

👉 将指定的文件读入到当前位置

👉 文件名可不带扩展名, 缺省为 `.tex`

👉 `\input` 命令可以出现在文档的任何部分

👉 `\input` 命令可以嵌套

👉 `\listfiles` → 放在导言区, 在屏幕和 log 文件中列出文件清单

👉 编译时, 读入所有文件, 再进行编译

# 文档的分割处理

## ❑ `\include` 命令: `\include{文件名}`

- ☞ 相当于 `\clearpage\input{文件名}\clearpage`
- ☞ 只能出现在正文部分
- ☞ `\include` 命令不可以嵌套
- ☞ 优点: 与 `\includeonly{文件清单}` 配合使用
  - 编译时, 只读入 `\includeonly` 指定的文件清单
  - 保持有关页面, 章节, 公式编号等相关信息
- ☞ `\includeonly` 只能放在导言区
- ☞ 若没有 `\includeonly` 命令, 则读入所有文件

- 1 document选项
- 2 页面版式
- 3 自定义页眉与页脚
- 4 目录表与图表目录
- 5 文档的分割处理
- 6 交叉引用与索引



# 交叉引用

❑ 交叉引用:  $\text{\LaTeX}$  的一个非常有用的功能

👉 章节, 表格, 插图, 公式, 页码, 参考文献等

👉  $\text{\LaTeX}$  自动跟踪所有的变动,  
保证所有交叉引用与索引记录的正确性

❑ `\label{标记}`: 设置标记

👉 `\label` 命令常出现在章节, 公式, 定理类环境, 枚举罗列, `table`, `figure` 中的 `\caption` 参数中.

❑ `\ref{标记}`: 显示标记所对应的对象的编号

❑ `\pageref{标记}`: 显示标记所在页面的页码

# 索引

❑ 索引: 在导言区加入

```
\usepackage{makeidx}  
\makeindex
```

👉 `\index{索引条目}` → 标记索引条目

👉 `\printindex` → 输出索引条目

👉 索引标题: `\indexname`

👉 索引按ASCII码排序 (中文索引排序问题见教材)

👉 多索引问题: `index` 宏包